Docket No.: 0826.1718

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re the Application of:

Yoshitake SHINKAI

Serial No. 09/817,288 Group Art Unit: 2152

Confirmation No. 7742

Filed: March 27, 2001 Examiner: Lesniewski, Victor D.

For: FILE REPLICATION SYSTEM, REPLICATION CONTROL METHOD, AND STORAGE MEDIUM

BRIEF IN SUPPORT OF APPEAL

Mail Stop Appeal Brief-Patents
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Final Office Action in the above-identified application, and pursuant to the Notice of Appeal filed on January 2, 2007, Applicants submit the present Brief with the fee of $500.00 set forth by 1.17(c). A Petition for an Extension of Time and the required fee of $450.00 requesting a one-month extension are concurrently filed herewith, extending the period for filing the Brief in Support of Appeal to May 2, 2007.

(I) Real Party In Interest

The real party in interest in the present appeal is the assignee Fujitsu Limited.

(II) Related Appeals and Interferences

The undersigned attorney, the appellant, and the assignee know of no related appeals or interferences which would be directly affected by or directly affect or have a bearing on the Board's decision in the present appeal.

05/02/2007 YPOLITE1 00000008 09817288
01 FC:1402 500.00 OP

(III) Status of Claims

Claims 1-8, 10-12, 14, 16-27, and 28 are currently pending. Claims 1-8, 10-12, 14, and 16-28 stand finally rejected and are appealed.

05/02/2007 YPOLITE1 00000008 09817288
02 FC:1252 450.00 OP

(IV) Status of Amendments

No amendments have been filed subsequent to the final rejection.

(V) Summary of Claimed Subject Matter

The present invention relates to a file replication technology for dynamically distributing file replications of a file to a plurality of computers so as to distribute the load of the system to improve system performance and to enhance system reliability.

Independent Claim 1

Independent claim 1 is directed to a file replication system having a plurality of nodes connected to a network wherein shared files are distributed to the nodes. For example, in at least one embodiment of the present invention, as illustrated in Figure 1 and as defined by claim 1, a node 1 includes a file 6 that is shared with another node. The node 1 also includes an IO Request Intercepting Portion 2 and a first Token Managing Portion, which corresponds to the Token Managing Portion 3 of Figure 1. *See* specification of the present invention, page 13, lines 9-15. *See also* FIG. 1 of the present invention. The system also includes a second node including a Second Token Managing Portion 13 of FIG. 4.

The Token Managing Portion 3 manages access requests for a shared file 6. The IO Request Intercepting Portion 2 asks the Token Managing Portion 3 to permit access to the shared file 6 in response to an access request for the shared file 6 by the node 1. When the Token Managing Portion 3 allows the access, the IO Request Intercepting Portion 2 accesses the shared file 6.

When a second node has update permission for the shared file 6, the Token Managing Portion 3 notifies the IO Request Intercepting Portion 2 of the second node having update permission for the shared file 6, in response to the access request by the second node. The second node having the update permission is then granted access to the shared file 6. As a result, each node 1 can access the data of a node that has the latest data. In addition, each node can access consistent data.

Independent Claim 10

Independent claim 10 is directed to a file replication system having a plurality of nodes connected to a network wherein shared files are distributed to the nodes. For example, in at least one embodiment of the present invention, as illustrated in Figure 1 and as defined by claim 10, a node 1 includes a file 6 that is shared with another node. The node 1 also includes an IO request intercepting means, which corresponds to the IO Request Intercepting Portion 2 and a

first token managing means, which corresponds to the Token Managing Portion 3. *See* specification of the present invention, page 13, lines 9-15. *See also* FIG. 1 of the present invention. The system also includes a second node including a second token managing means, which corresponds to the token managing portion 13 of FIG. 4. The components recited in claim 10 function as explained in the section regarding Independent claim 1.

Independent Claim 2

Independent claim 2 is directed to a node in a file replication system having a plurality of nodes connected to a network wherein shared files are distributed to the nodes. In at least one embodiment, the node includes a Token Managing Portion 3 managing an access request for a file and an IO Request Intercepting Portion 2. The IO Request Intercepting Portion 2 asks the Token Managing Portion 3 to permit access to the shared file 6, in response to an access request for the shared file 6 in the node itself. When the Token Managing Portion 3 permits the access, the IO Request Intercepting Portion 2 accesses the shared file 6. *See* specification of the present invention, page 13, lines 18-24. *See also* FIG. 1 of the present invention.

When another node has update permission for the shared file 6, the Token Managing Portion 3 notifies the IO Request Intercepting Portion 2 of the node that has the update permission for the shared file 6, in response to the access request. When the IO Request Intercepting Portion 2 cannot acquire the access permission, it asks the node that has the update permission to access the shared file 6.

Independent Claim 11

Independent claim 11 is directed to a node in a file replication system having a plurality of nodes connected to a network wherein shared files are distributed to the nodes. In at least one embodiment, the node includes a Token Managing Means managing an access request for a file and an IO request intercepting Means. The Token Managing Means corresponds to the Token Managing Portion 3. The IO Request Intercepting Means corresponds to the IO Request Intercepting Portion 2. The components of independent claim 11 function in the manner described in the section regarding "Independent Claim 2. *See* specification of the present invention, page 13, lines 18-24. *See also* FIG. 1 of the present invention.

Independent Claim 8

Independent claim 8 is directed to a node in a file replication system having a plurality of nodes connected to a network wherein shared files are distributed to the nodes. In at least one embodiment, the node includes a Token Managing Portion 3 managing an access request for a

file and an IO Request Intercepting Portion 2. As defined by claim 8, the Token Managing Portion 3 asks another node to acquire an access permission for a file against an access request for the file in the node. The IO Request Intercepting Portion 2 accepts an access request for a file in the node and asks the Token Managing Portion 3 to acquire the access permission for the file against the access request to the file in the node. When no other node has update permission for the file, the token managing portion gives access permission for the file and otherwise notifies the IO Request Intercepting Portion of another node that has the update permission for the file. If the IO Request Intercepting Portion 2 is capable of acquiring the access permission , the IO Request Intercepting Portion 2 accesses the file. If the Token Managing Portion 3 is not capable of acquiring access permission for the file, the IO Request Intercepting Portion 2 asks the other node that has update permission for the file to access the file according to the access request. *See* specification of the present invention, page 13, lines 18-24. *See also* FIG. 1 of the present invention.

Independent Claim 12

Independent claim 12 is directed to a node connected to at least one other node through a network wherein every node includes a copy of files synchronized with files of other nodes for high availability. As defined by claim 12, the node includes a token managing means corresponding to the Token Managing Portion 3. The node further includes an IO Request Intercepting Means corresponding to an IO Request Intercepting Portion 2. The components of independent claim 12 function in the manner described in the section regarding "Independent Claim 8." *See* specification of the present invention, page 13, lines 18-24. *See also* FIG. 1 of the present invention.

Independent Claims 14 and 28

Independent claims 14 and 28 are directed to a file replication control method and computer-readable portable storage medium, respectively, for a system having a plurality of nodes connected to a network. According to the method, in response to an access request for a shared file in a particular node, an IO Request Intercepting Portion 2 of the node accesses the shared file 6.

When another node has update permission for the shared file 6, the token managing portion 3 notifies the IO Request Intercepting Portion 2 of the node that has the update permission for the shared file 6, in response to the access request. In the event that the IO Request Intercepting portion 2 cannot acquire access permission, it asks the node that has the update permission to access the shared file 6. As a result, each node 1 can access the data of

a node that has the latest data. Each node can also access consistent data. *See* specification of the present invention, page 13, lines 12-24. *See also* specification of the present invention, page 13, line 25 – page 14, line 8. *See also* Fig. 1.

Independent Claim 27

Independent claim 27 is directed to a file replication method for a system having a plurality of nodes connected to a network. According to the method, when a user program, for example, of each node issues an access request for a file of the object group, the node issues a read/write token acquisition request to the node A. Unless the node A has already given a write token to another node, the node A gives the token to the requesting node. When the node A has already given the write token to another node, the node A notifies the requesting node of a node that has the write token. When the requesting node receives a token acquisition failure message, the requesting node asks the notified node to process a read/write request for the file. As a result, the node having the write token processes such requests so that the order of write operation to the file is maintained. As illustrated in Fig. 3A, when the nodes B and C issue read requests (reference requests) and the node D issues a write request (update requests) and the node D issues a write request (update request), the node A notifies each node that the node E has the write token , along with a token acquisition failure message in response to token acquisition requests therefrom. *See* FIG. 3A.

(VI) Grounds of Rejection to be Reviewed on Appeal

A. Claims 1-3, 8, 10-12, 14, 23, 25, 27, and 28 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over U.S. Patent Number 5,964,886, issued to Slaughter *et al.* (hereinafter referred to as Slaughter) in view of U.S. Patent Number 5,634,122, issued to Loucks *et al.* (hereinafter referred to as Loucks).

B. Claims 4-7, 16-22, 24, and 26 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Slaughter in view of Loucks, and further in view of U.S. Patent Number 5,515,537, issued to Tavares (hereinafter referred to as Tavares).

(VII) Argument

A. Claims 1 and 10 stand rejected as being allegedly unpatentable over Slaughter in view of Loucks. Claims 1 and 10 stand or fall together as a group.

1. Background of the References on which the Rejection is Based

Slaughter is directed to a distributed computer system including a plurality of nodes of a cluster. Each node of the cluster has access to each storage device of the cluster. When client 312A accesses data from a storage device, it sends a data access request to Net Disk Drive 318A. Net disk drive 318A, based on the mapping and current membership information, determines to which node to convey the data access request. Net disk drive 318A may route the data access request to the primary node if the primary node is active. Alternatively, if the primary node is not active, then the Net Disk drive 318A may route the data access request to the secondary node.

Loucks is directed to a system and method for controlling access to shared resources in a distributed computer system. Access to shared resources is controlled by a local authorization token manager. According to Loucks, its system and method can be used by server machines for distributed file systems to synchronize access to files between protocol exporters of different distributed file systems and local processes. The distributed file system protocol experts can then synchronize access by its clients to exported volumes. The client machines can also synchronize access to cached files between different processes on the client machine

2. Relevant Law

To establish a *prima facie* case of obviousness, one of the three basic criteria that must be met is that the reference must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the references, and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). See MPEP § 2143 - § 2143.03 for decisions pertinent to each of these criteria.

The initial burden is on the examiner to provide some suggestion of the desirability of doing what the inventor has done. "To support the conclusion that the claimed invention is directed to obvious subject matter, either the references must expressly or impliedly suggest the claimed invention or the examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references." *Ex parte Clapp*, 227 USPQ 972, 973 (Bd. Pat. App. & Inter. 1985). See MPEP § 2144 - § 2144.09 for examples of reasoning supporting obviousness rejections.

3. Application of the Relevant Law

Applicants respectfully submit that neither Slaughter nor Loucks, alone or in combination, teaches or suggests a first node that includes an I/O request intercepting portion accepting

access to a file wherein the access occurs in the first node when the IO request intercepting portion is capable of acquiring access permission, as indicated by the language of claim 1.

In contrast to the present invention, In Slaughter, the net disk drive 318A does not include an I/O request intercepting portion that accepts access to a file. Rather, after receiving a data access request, the net disk drive 318A simply forwards the request to a primary node if the primary node is active. Alternatively, if the primary node is not active, then the net disk drive 318A may route the data access request to a secondary node. That is, in Slaughter, file access does not actually occur in the net disk driver 318A, as the net disk drive 318A forwards the data access request to either a primary node or a secondary node. *See* Slaughter, column 9, line 29 – line 34.

On page 3 of the Office Action, the Examiner stated that "[w]hen the primary node is not active (meaning "not capable of acquiring the access permission"), the data access request is routed to a secondary node which is active (meaning "asking the permitted node to access to the file"). *See* Office Action, page 3, item 8.

Applicants respectfully submit that in contrast to the present invention, in Slaughter, the net disk driver 318A does not ask a permitted node that has update permission for the file to access the file. In fact, Slaughter simply transmits the access request to another node, without regard to what entity has update permission for a file. The other node is not necessarily the node that has access permission. That is, the node to which Slaughter transmits the data access request may then forward the data access request to yet another node.

For example, the netdisk driver 318A may forward the data access request to NM 320A and NM 320 A will then convey the request to disk driver 326A, which, in turn accesses the storage device. Therefore, NM 320A, to which the netdisk driver 318A transmitted the data access request does not have access permission, as the disk driver 326A has the permission. Therefore, the netdisk driver 318A does not ask a permitted node to access the file.

Loucks simply discloses file sharing via an exported volume using the exporting distributed file system protocol. Therefore, Loucks does not cure the deficiencies of Slaughter.

In light of the foregoing, claims 1 and 10 are patentable over the references.

B. Claims 2, 8, 11, and 12 stand rejected as being allegedly unpatentable over Slaughter in view of Loucks. Claims 2, 8, 11, and 12 stand or fall together as a group.

1. <u>Relevant Law</u>

To establish a *prima facie* case of obviousness, one of the three basic criteria that must

be met is that the reference must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the references, and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). See MPEP § 2143 - § 2143.03 for decisions pertinent to each of these criteria.

The initial burden is on the examiner to provide some suggestion of the desirability of doing what the inventor has done. "To support the conclusion that the claimed invention is directed to obvious subject matter, either the references must expressly or impliedly suggest the claimed invention or the examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references." *Ex parte Clapp*, 227 USPQ 972, 973 (Bd. Pat. App. & Inter. 1985). See MPEP § 2144 - § 2144.09 for examples of reasoning supporting obviousness rejections.

2. Application of the Relevant Law

Applicants respectfully submit that neither Slaughter nor Loucks, alone or in combination, teaches or suggests a token managing portion ". . . asking the other node that has the update permission for the file to access the file according to the access request," as recited in claim 2, for example.

Applicants respectfully submit that in contrast to the present invention, in Slaughter, the net disk driver 318A does not ask another node that has update permission for a file to access the file according to an access request.

Rather, Slaughter simply transmits the access request to another node, without regard to what entity has update permission for a file. The other node is not necessarily the node that has access permission. That is, the node to which Slaughter transmits the data access request may then forward the data access request to yet another node having access permission.

For example, the netdisk driver 318A may forward the data access request to NM 320A and NM 320 A will then convey the request to disk driver 326A, which, in turn accesses the storage device. Therefore, NM 320A, to which the netdisk driver 318A transmitted the data access request does not have access permission, as the disk driver 326A has the permission. Therefore, the netdisk driver 318A does not ask another node that has update permission for a file to access the file.

Loucks simply discloses file sharing via an exported volume using the exporting distributed file system protocol. Therefore, Loucks does not cure the deficiencies of Slaughter.

In addition, Slaughter does not disclose or suggest, "said token managing portion giving access permission when no other node has update permission for the file," as recited in claim 8 of the present invention. Slaughter's netdisk driver 318A does not include a token managing portion that gives access permission when no other node has permission. Rather, Slaughter specifically indicates that the netdisk driver 326A forwards the data access request to another node. Thus, access is not given in the netdisk driver 318A, as the node to which the request is forwarded has the permission, that is, the destination node.

As Loucks does not cure the deficiencies of Slaughter, claims 2, 8, 11, and 12 are patentable over the combination of references.

C. Claims 14, 27, and 28 stand rejected as being allegedly unpatentable over Slaughter in view of Loucks. Claims 14, 27, and 28 stand or fall together as a group.

1. **Relevant Law**

To establish a *prima facie* case of obviousness, one of the three basic criteria that must be met is that the reference must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the references, and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991). See MPEP § 2143 - § 2143.03 for decisions pertinent to each of these criteria.

The initial burden is on the examiner to provide some suggestion of the desirability of doing what the inventor has done. "To support the conclusion that the claimed invention is directed to obvious subject matter, either the references must expressly or impliedly suggest the claimed invention or the examiner must present a convincing line of reasoning as to why the artisan would have found the claimed invention to have been obvious in light of the teachings of the references." *Ex parte Clapp*, 227 USPQ 972, 973 (Bd. Pat. App. & Inter. 1985). See MPEP § 2144 - § 2144.09 for examples of reasoning supporting obviousness rejections.

2. **Application of the Relevant Law**

Applicants respectfully submit that neither Slaughter nor Loucks, alone or in combination, teaches or suggests "causing an access requesting node to access a file of the access requesting node itself when the access requesting node has the latest data of the file and has or is able to obtain access permission from another node having update permission for the file," as recited in claim 14, for example.

Applicants respectfully submit that in contrast to the present invention, in Slaughter, the

node from which the data access request originated does not access a file. Rather, another node to which the data access request is sent, that is, the destination node, actually performs the access operation. *See* Slaughter, column 9, lines 35-39.

Loucks simply discloses file sharing via an exported volume using the exporting distributed file system protocol. Therefore, Loucks does not cure the deficiencies of Slaughter.

In light of the foregoing, claims 14, 27, and 28 are patentable over the references.

Conclusion

Applicants respectfully submit that the Examiner has not established a prima facie case of obviousness by preponderance of the evidence for the relevant claims. Reversal of the rejection is, therefore, requested.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 5-1-07

By: _____
Reginald D. Lucas
Registration No. 46,883

1201 New York Avenue, NW, Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

VIII. Claims Appendix

1. (previously presented) A file replication system having a plurality of nodes connected to a network, files being distributed to the nodes, wherein

a first node of the nodes comprises:

a first token managing portion giving access permission for a file within the first node when no other node has update permission and otherwise issuing a notification of a permitted node that has update permission for the file in response to an access request in the first node, and

an IO request intercepting portion accepting an access to the file, the access taking place in the first node when said IO request intercepting portion is capable of acquiring the access permission, asking said first token managing portion to acquire the access permission against the access request, and asking the permitted node that has update permission for the file to access to the file when said first token managing portion is not capable of acquiring the access permission, and

a second node comprises a second token managing portion notifying a requesting node that requests the access permission for the file of the permitted node that has the update permission for the file as a response message.

2. (previously presented) A node, connected to at least one other node through a network, every node having a copy of files synchronized with files of other nodes for high availability and high performance to provide a replicated file system, comprising:

a token managing portion managing an access request for a file; and

an IO request intercepting portion asking said token managing portion to acquire access permission for the file against an access request to the file in said node, said token managing portion giving access permission when no other node has update permission for the file and said token managing portion notifying said IO request intercepting portion of another node that has the update permission when the other node has the update permission for the file, in response to the access request of said IO request intercepting portion, said IO request intercepting portion accessing the file in said node when the IO request intercepting portion is capable of acquiring the access permission and said IO request intercepting portion asking the other node that has the update permission to access the file instead of accessing the file in said node when said IO request intercepting portion is not capable of acquiring the access permission.

3. (previously presented) The node according to claim 2, further comprising:

a system structure managing portion performing a restoration process of data of a file of the node when it is newly joined to a system,

wherein while said system structure managing portion is restoring the file, when an access request for the file takes place in the node, said IO request intercepting portion asks another node that shares the file to access the file.

4. (previously presented) The node according to claim 2, further comprising:

a changed data notifying portion propagating an updated content of the file to other node along with information that represents a dependent relationship with another update; and

a received data processing portion reflecting the updated content to the file while assuring an order of the update based on the dependency relationship.

5. (previously presented) The node according to claim 4, further comprising:

a system state information portion storing information about propagation mode of an updated content for each of at least one file,

wherein said changed data notifying portion propagates the update content based on information queued in said system information portion.

6. (previously presented) The node according to claim 5, wherein the propagation mode is one of a synchronous mode in which it is assured that the updated content is propagated to all the nodes that share the file, a semi-synchronous mode in which it is assured that the updated content is propagated to the majority of nodes that share the file, and an asynchronous mode in which it is not acknowledged that the updated content is propagated to the nodes that share the file.

7. (previously presented) The node according to claim 4, wherein said system state information storing portion keeps information about each node that shares at least one file for each file.

8. (previously presented) A node, connected to at least one other node through a network, every node having a copy of files synchronized with files of other nodes for high availability and high performance to provide a replicated file system, comprising:

a token managing portion asking another node to acquire an access permission for a file

against an access request for the file in said node; and

an IO request intercepting portion accepting an access request for a file in said node, asking said token managing portion to acquire the access permission for the file against the access request to the file in said node, said token managing portion giving access permission when no other node has update permission for the file and otherwise notifying said IO request intercepting portion of another node that has the update permission for the file, said IO request intercepting portion accessing the file in said node when the IO request intercepting portion is capable of acquiring the access permission and asking the other node that has the update permission for the file to access the file according to the access request instead of accessing the file in said node when said token managing portion is not capable of acquiring the access permission for the file.

9. (cancelled)

10. (previously presented) A file replication system having a plurality of nodes connected to a network, files being distributed to the nodes, wherein

a first node of the nodes comprises:

first token managing means for giving access permission for a file within the first node when no other node has update permission and otherwise issuing a notification of a permitted node that has update permission for the file in response to an access request in the first node, and

IO request intercepting means for accepting an access to the file, the access taking place in the first node when said IO request intercepting portion is capable of acquiring the access permission, asking said first token managing means to acquire the access permission against the access request, and asking the permitted node that has update permission for the file to access to the file when said first token managing means is not capable of acquiring the access permission, and

a second node comprises second token managing means for notifying a requesting node that requests the access permission for the file of the permitted node that has the update permission for the file as a response message.

11. (previously presented) A node, connected to at least one other node through a network, every node having a copy of files synchronized with files of other nodes for high availability and high performance to provide a replicated file system, comprising:

token managing means for managing an access request for a file; and

IO request intercepting means for asking said token managing means to acquire an access permission for the file in response to an access request to the file in said node, said token managing means giving access permission when no other node has update permission for the file and said token managing portion notifying said IO request intercepting means of another node that has the update permission when the other node has the update permission for the file, in response to the access request of said IO request intercepting means, said IO request intercepting portion accessing the file in said node when the IO request intercepting portion is capable of acquiring the access permission and said IO request intercepting means asking the other node that has the update permission to access the file instead of accessing the file in said node when said IO request intercepting means is not capable of acquiring the access permission.

12. (previously presented) A node, connected to at least one other node through a network, every node having a copy of files synchronized with files of other nodes for high availability and high performance to provide a replicated file system, comprising:

token managing means for asking another node to acquire an access permission for a file against an access request for the file in said node; and

IO request intercepting means for accepting an access request for a file in said node, asking said token managing means to acquire the access permission for the file against the access request to the file in said node, said token managing portion giving access permission when no other node has update permission for the file and otherwise notifying said IO request intercepting portion of another node that has the update permission for the file, said IO request intercepting portion accessing the file in said node when the IO request intercepting portion is capable of acquiring the access permission and asking the other node that has the update permission for the file to access the file according to the access request instead of accessing the file in said node when said token managing means is not capable of acquiring the access permission for the file.

13. (cancelled)

14. (previously presented) A file replication control method for a system having a plurality of nodes connected to a network, files being distributed to the nodes, comprising:

causing an access requesting node to access a file of the access requesting node itself

when the access requesting node has the latest data of the file and has or is able to obtain access permission from another node having update permission for the file; and

asking the other node to access the file when the other node has the update permission for the file which is given to only one node at a time.

15. (cancelled)

16. (previously presented) The file replication control method according to claim 14, wherein the other node that has the update permission releases the update permission after an update that has a dependent relationship with the update performed at the other node has been propagated to all the nodes.

17. (previously presented) The file replication control method according to claim 14, wherein said method further comprises:

the other node that has updated the file asynchronously propagating an updated content to the other nodes; and

causing the other node that has updated the file to process an access request that takes place in the access requesting node while the updated content is being propagated.

18. (original) The file replication control method according to claim 17, wherein the updated content is reflected in such a manner that order thereof is assured.

19. (original) The file replication control method according to claim 18, wherein a dependency information that represents order of other updates to be propagated to the other node along with the updated content.

20. (previously presented) The file replication control method according to claim 19, wherein a node that has received the updated content to reflect the updated content on a file of the node itself after receiving a previous updated content based on the dependency information.

21. (previously presented) The file replication control method according to claim 14, wherein a propagation mode of an updated content is designated for each of at least one file.

22. (previously presented) The file replication control method according to claim 14,

wherein a node to which an updated content is propagated is designated for each of at least one file.

23. (previously presented) The file replication control method according to claim 14, further comprising:

restoring data of a file of a newly joined node; and

operating a user program before data of the file is completely restored.

24. (previously presented) The file replication control method according to claim 23, wherein restored data is transmitted in such a manner that order of update requests for the file is assured.

25. (previously presented) The file replication control method according to claim 23, wherein the node asks another node that shares the file to perform a process for an access request for the file when the access request takes place in the node itself before data is completely restored.

26. (previously presented) The file replication control method according to claim 14, wherein a node that has performed a systematic stop in which nodes that share a file are synchronously stopped to store a systematic stop state and the node synchronously resumes a process for the file without restoring data of the file.

27. (previously presented) A file replication method for a system having a plurality of nodes connected to a network, files being distributed to the nodes, comprising:

causing a first node to request a token for accessing a file;

causing the first node to access the file when the first node has the latest data of the file and is able to obtain the token for accessing the file from another node having update permission for the file which is given to only one node at a time;

notifying the first node of a second node that has the token when the first node is not capable of acquiring the token; and

causing the first node to ask the second node to access the file when the first node is notified that the first node is not capable of acquiring the token.

28. (previously presented) A computer-readable portable storage medium, when being

used by a computer that composes a node connected to other nodes through a network in a file replication system, on which is recorded a program for causing the computer to execute a process, said process comprising:

when the node accesses a file and a node itself has the latest data of the file and has or is able to obtain access permission from another node having update permission for the file, causing the node itself to access the file of the node itself; and

when another node has the update permission for the file which is given to only one node at a time, causing the node itself to ask the other node to access the file.

IX. Evidence Appendix

Not applicable.

X. Related Proceedings Appendix

Not applicable.